Assignment 2: Your Best Software Project

For this assignment, you'll be developing:

- A plan on where you'll think through where you are and how to initiate continuous improvement across the four jobs of software development: Learning, Deciding, Building and Managing

Assignment Instructions
Think through how you will (or how you currently) approach the four jobs of software in your company, using the charts below to outline your plan.

Note: As you develop your submission, please be sure to organize your responses underneath the various questions (1a, 2c, etc.) or in a chart so it's easy for your reviewer to understand what pertains to each.

1. Describe your "learning machine" and how you'll create a virtuous cycle of learning that will continually improve your team's understanding of the user.

Items to consider	Your plan for Learning
a. What vehicles (observations, metrics, tests, etc.) will you and your team use to make learning a regular part of your agile iterations?	We will use the regular design sprints to ensure that the team is mission driven on delivering value to the customers instead of focusing on churning outputs in the form of code. We will use INVEST- able user stories. Employ customer value-based metrics to measure performance of the team and the sprint and will borrow continuous testing regime from XP to make sure we don't accumulate bugs and errors since we don't have dedicated testers/QA in the team.
b. What metrics will you use to measure outcomes?	1. Number of customer interactions (an interaction means a request from app developer to our system to query about a customer's balance) 2. Number of customer transactions (a transaction is when the customer interacts with the in-app message and the developer sends a request to check eligible offers for the customer) 3. Number of customer purchases (how many purchases customers made)
c. Who will take the lead on designing that learning? Who will take the lead on sharing it with the team in a way that drives strong, interdisciplinary discussions and links to the job of Deciding?	The Product Manager (myself) as an advocate of the customer and business will take the lead on designing and sharing the learning with the team and will also use the learning for deciding.

d. How often and in what meetings/workshops will you deliver the above?	During every sprint planning, backlog grooming and sprint demo
e. How will you present your user stories and related items (storyboards, story maps, etc.) to drive the best possible discussions in your sprint planning meetings?	We will use the whiteboard next to the desks where the team sits to place the user stories and storyboards/maps always. We will also use the soft-copy on Google Docs so that it’s readily available for the team members who would like to work remotely some days and also use it for sprint planning meetings.
f. What about after the sprint starts? How will you work to tilt the working environment towards thinking about what makes sense vs. just creating output--those ‘blue button’ moments?	The persona map and storyboard remains visible to the team always which will drive the conversation more towards what value can we create for the customer. We will also have the metrics on display on big TV screen where the live dashboard will be visible at all times for the entire team to focus on data-driven outcomes vs opinion based discussions.
g. What are the challenges and focal points you see for linking to the job of Deciding and creating a culture of experimentation?	The team is currently not used to culture of experimentation so changing the way of work where we don’t consider a user story as done unless it has been “validated” is going to be challenging since some of the results may not come soon enough before the next sprint starts. It will then be challenging to keep track of the open user stories from previous sprints that haven’t been validated while we accumulate the new ones to work on.

2. Describe how you’ll decide what to build, when.

Items to consider	Your plan for Deciding
a. How long will your iterations be--and why? What might be the impact of longer iterations? Shorter?	We will have two week sprints because it’s a good enough time to make some tangible progress from previous sprint while short enough to adapt to changes in priorities and customer needs. If we make the sprint longer than this it might result in more “hard stops” to respond to critical customer needs. A sprint shorter than two weeks will be too short to make a significant enough progress sprint over sprint and we will end up spending more time in planning, demo and retro meetings rather than building software.
b. How will you groom the product backlog to increase the quality of your inputs and prioritize tasks? What inputs and what people will be most important to your pre-sprint backlog grooming sessions? Why?	I will document the meeting notes from stakeholder/customer meetings. This product is a B2B so we will have to give a lot of updates to the business units involved and our partner developers who are the customers as well. So, we will use those inputs to bring to the grooming sessions. The entire team will be part of the sessions to ensure everyone can pitch in with their inputs since luckily engineers can bring added perspective since it’s a product for developers by developers.

c. What is most critical to manage flow across design, development, and testing? For example, how will you make sure testing isn’t backloaded to the very end of the sprint?	Since we are going to use “validated” before “done” on our Kanban board for each user story, this will ensure that every story goes through design, development and testing phase before deployment for experimentation. Every user story is an experiment.
d. Which practices from XP, scrum, and kanban will you use to make the job of deciding more effective?	We will focus on customer value outcomes/impact over velocity, testable hypotheses with limited blast radius and frequent iterations to ensure the prioritzatin and deciding is easier. The Kanban board will be designed where “validated” comes before “done” and borrow unit testing from XP.
e. How will you evaluate the quality of your decisions and think about how to improve them?	The quality of decisions will be purely based on data. If our decisions are leading to better customer outcomes measured by the metrics, we are on the right track. If not, we will use the retrospectives and do qualitative user research to figure out what needs to change in our approach.

3. Describe how you’ll build (and test) software.

Items to consider	**Your plan for Building**
a. How will you support the development and testing team? If it’s an existing team, which practices (from XP, scrum, and kanban) do you use? Why those? How well do you think those are working and do you think there are any changes worth considering? If it’s a new team, which practices (from XP, scrum, and kanban) will you discuss with the team? Why those?	We will use test-first development with continuous integration and focus on validated learning
b. How will you frame the ‘win’ on the practices you suggest for developers? Testers? Designers? (Choose whichever roles you currently have on the team.)	We only have developers in the team since our product is a set of APIs. The team only wins when the customer “wins” and that will be substantiated by metrics specified above. To measure short term wins, we will measure the number of validated learnings/experiments as a measure to build the culture of experimentation and more focus on releasing a testable product often.
c. How will you discuss the best way to run testing and decide who on the team will do what?	We will borrow test-first development from XP and every developer will be responsible for testing.

d. How would you like to evolve the process of testing and deployment over time? What do you see as the first few steps?	We would like to explore solutions to automate the testing process.
e. How will you create slack to allow the team to avoid or reduce technical debt?	XP is a great way to reduce technical debt.

4. Describe how you will manage software development.

Items to consider	Your plan for Managing
a. What are the top three things you can do in your role to foster and contribute to a self-organizing team?	Principles > Rules Cross-pollination > standardization Agile > Scrum I will shield the team from outside pressure but at the same time bring the inputs to facilitate narrative collaboration. This will ensure high alignment yet high autonomy.
b. What is the role of the retrospective for you? What agenda will you use? How will you tie the results of the retrospective back to the job of learning?	Retrospectives are a great way for the team to reflect and focus on continuous learning and improvement. The agenda will be: What worked What needs to change